REMARKS

In the Office Action dated May 6, 2004, claims 1-4, 13-24, and 33-35 were rejected under 35 U.S.C. § 103 over Urhan, "XJoin: Getting Fast Answers from Slow and Bursty Networks," CS-TR-3994 (Urhan), in view of U.S. Patent No. 5,873,074 (Kashyap); and claims 5-12 and 25-32 were rejected under § 103 over Urhan in view of Kashyap and DeWitt, "Parallel Sorting on a Shared-Nothing Architecture Using Probabilistic Splitting," PDIS (DeWitt).

Applicant respectfully submits that independent claims 1, 13, and 23 are not obvious over the asserted combination of Urhan and Kashyap. The Office Action has failed to establish that there is any motivation or suggestion to combine the teachings of Urhan and Kashyap. In fact, a careful reading of the references will reveal that there is no such motivation or suggestion to combine, as the proposed modification of Kashyap based on the teachings of Urhan being urged by the Office Action would render the system of Kashyap inoperative for its intended purpose. "If a proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)." MPEP § 2143 (8th ed., Rev. 2) at 2100-131.

For efficient parallel processing, Kashyap states that each bucket in the hash table of each instance of a hash-join operator should ideally receive approximately the same amount of data. Kashyap, 2:3-5. Kashyap notes, however, that such an ideal distribution may not occur because the distribution of data values in "build" and "probe" tables is skewed or unevenly distributed. The hash-join mechanism described in Kashyap is designed to address the skew (uneven distribution) of either the "build" table or "probe" table.

To this end, Kashyap describes a partitioning technique for distributing records retrieved from a table among a first level of instances of a hash-join operator 28 (Kashyap, 6:15-19). However, to address the skew (uneven distribution) problem, the hash-join operator 28 of Kashyap must wait until *all* records of a build table 20 have been distributed into buckets of hash tables before any hash-join operation is performed. *See* Kashyap, Figures 3, 4A-4B, columns 5-12. The distributions of records of the build table 20 and of the probe table 22 are performed in sequence--the build table 20 is first distributed into buckets of a hash table, and after *all* records of the build table 20 have been distributed, the probe table 22 is distributed. *See* Kashyap, 9:12-24. Such a sequential distribution of records of the build table 20 and the probe table 22 of Kashyap is required to enable skew detection. *See* Kashyap, 6:57-59. As explained by Kashyap, skew is detected if a bucket receives too many records of the build table 20. When skew is detected, records of the build table 20 are re-routed to a second level of instances of the hash-join operator 32 (rather than to the first level of instances of the hash-join operator 28). *See* Kashyap, 7:42-52. Modifying Kashyap to enable incremental production of results during distribution of the records of the build and probe tables, as proposed by the Office Action, would render the Kashyap hash-join mechanism inoperative for its intended purpose, namely to enable skew or uneven distribution detection. Therefore, there is no motivation or suggestion to combine Kashyap and Urhan in the manner proposed by the Office Action.

Moreover, one objective of the XJoin approach as described in Urhan is based on the principle of producing results incrementally as they become available so that early delivery of initial answers can be achieved. A further objective of the XJoin approach of Urhan is that the approach allows progress to be made even when one or more sources experience delays. As discussed by Urhan, when a tuple arrives at one of its inputs, the tuple is inserted into a hash table for that input and then immediately used to probe the hash table of the other input. Urhan, p. 4. "Thus, even if one source becomes temporarily blocked, the operator can still produce results." *Id.* This teaching of Urhan is inconsistent with the technique employed in Kashyap, where all records of a first table (the build table 20) must first be distributed to buckets among first-level instances and second-level instances of a hash-join operator 28 *prior* to the distribution of records of a second table (probe table 22). As discussed above, Kashyap performs this sequential distribution of records from two tables for handling skew in the distribution of records. If skew occurs, such as when a bucket is receiving too great a share of build records, Kashyap teaches that subsequent records are re-routed to a second level of instances of a hash-join operator (to a "spilled" bucket). Kashyap, 7:5-56. Following the distribution of all build records (from the first table or build table 20), distribution of the probe records (from the second table or probe table 22) is mapped according to whether spilling has occurred. Kashyap, 9:32-49. In other words, distribution of records of the second table are dependent upon distribution of records of the first table (based on whether spilling has occurred or not). Therefore, the objective sought by the XJoin technique of Urhan would be defeated by the technique employed in

Kashyap, since early production of answers would not be achieved if the Urhan join system has to wait for all records of one table to be distributed prior to distribution of records from a second table. Thus, combining the teachings of Kashyap into Urhan would also defeat the intended purpose of Urhan. Therefore, no motivation or suggestion exists to combine Urhan and Kashyap for this further reason. The obviousness rejection against claim 1 is therefore defective.

In addition, the obviousness rejection is defective for the additional reason that even if Urhan and Kashyap can be properly combined, the hypothetical combination of references fails to teach or suggest *all* elements of claim 1. *See* MPEP § 2143 at 2100-129. As conceded by the Office Action, Urhan fails to disclose redistributing first and second tuples to plural nodes according to a partitioning. However, contrary to the assertion made in the Office Action, Urhan also fails to disclose hash joining the first and second tuples to produce result tuples as the first and second tuples *are being redistributed to the plural nodes.*

Although the Office Action cited Kashyap as teaching the redistributing of first and second tuples to plural nodes according to the partitioning, the Office Action fails to explain how either Urhan or Kashyap teaches or suggests hash joining first and second tuples to produce result tuples *as the first and second tuples are being redistributed to plural nodes.* Urhan cannot perform such a hash joining task because Urhan does not disclose the redistribution of tuples to plural nodes. On the other hand, because Kashyap teaches that all records of the build table 20 (a first table) are distributed prior to distribution of the probe table 22 (the second table), *see* Kashyap 9:12-23, Kashyap cannot possibly teach or suggest hash joining first and second tuples to produce result tables *as the first and second tuples are being redistributed to the plural nodes.*

Therefore, even if Urhan and Kashyap can be properly combined, the hypothetical combination of the references fails to teach or suggest each and every element of claim 1. The obviousness rejection is defective for this additional reason.

Independent claim 13 is also not disclosed or suggested by Urhan and Kashyap. Claim 13 recites a database system that includes a plurality of nodes and instructions for enabling the database system to store first tuples in a first table distributed across the plurality of nodes, store second tuples in a second table distributed across the plurality of nodes, partition the first and second tuples into plural portions, redistribute the first and second tuples to the plurality of nodes according to the partitioning, and hash join the first and second tuples to produce result tuples as the first and second tuples are being redistributed to the plurality of nodes. As explained above, the asserted combination of Urhan and Kashyap does not disclose or suggest the hash joining of first and second tuples of first and second tables to produce result tuples *as the first and second tuples are being redistributed to the plurality of nodes.* The obviousness rejection against claim 13 is defective for at least the following reasons: (1) there is no motivation or suggestion to combine the Urhan and Kashyap to achieve the claimed subject matter; and (2) the hypothetical combination of Urhan and Kashyap fails to disclose or suggest *all* elements of claim 13.

Independent claim 23 is similarly allowable over the asserted combination of Urhan and Kashyap. Dependent claims are allowable for at least the same reasons as corresponding independent claims.

Dependent claims 5-12 and 25-32 were rejected as being obvious over an asserted combination of three references: Urhan, Kashyap, and DeWitt. Because no motivation or suggestion exists to combine the teachings of Urhan and Kashyap, the three-way combination of Urhan, Kashyap, and DeWitt also fails. Furthermore, because the hypothetical combination of Urhan and Kashyap fails to teach or suggest all elements of base claims 1, 13, and 23, the hypothetical combination of Urhan, Kashyap, and DeWitt would also fail to teach or suggest all elements of the dependent claims. Therefore, allowance of all claims is respectfully requested.

The Commissioner is authorized to charge any additional fees, including extension of time fees, and/or credit any overpayment to Deposit Account No. 50-1673 (9558).

Respectfully submitted,

Date: 7-6-04

Dan C. Hu, Reg. No. 40,025
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024
713/468-8880 [Ph]
713/468-8883 [Fax]